
C ++ C++ C++
C ++ C++ C++ C++ C ++ C++ C+
C++ C ++ C++ C++ C
تعلّم أن
تبرمج بـ ..
C++
المؤلف
خالد بن عبدالرحمن الحصيّن
كلية الحاسب – جامعة الامام

المقدمة

الحمد لله رب العالمين والصلاة والسلام على أشرف الأنبياء والمرسلين نبينا محمد وعلى آله وصحبه أجمعين وبعد :

فها قد تم ولله الحمد جمع الدورة التي نظّمت في شهر رمضان المبارك عام ألف وأربع مائة وتسع وعشرين.

وقد ألقيت الدورة في منتدى عشاق البرمجة التابعة لجامعة الإمام محمد بن سعود الإسلامية :

http://www.imam1.com/vb/forumdisplay.php?f=102

رابط الدورة:

http://www.imam1.com/vb/showthread.php?t=45042

وقد تم تقسيم هذا الكتاب إلى أربعة فصول وملحق خاص بحلول الواجبات والمشروع .

إقتصرت الدروس على الأساس ولم ندخل في التفاصيل .

وأحب في الختام أن أشكر والديّ اللذيّن سانداني على إنجاز هذا المشروع وكذالك أحب أن أشكر أخي عبدالله الذي ساندني في إعداد هذه الدورة، وكذالك أخي في الإسلام الذي أحبه في الله برقاوي جزاه الله عني كل خير حيث أشرف على الدورة وقد أضاف إضافات كثيرة في الدروس بل إن الفصل الأخير هو الذي كتبه وقد نبه على الكثير من الأخطاء وتعب في إعداد الدروس .

وكذالك أحب أن أشكر كلّ من ساند في إعداد هذه الدورة وكل من دخل الدورة وساهم في إنجاحها. وأسأل الله سبحانه أن ينفع بهذا الكتاب الإسلام والمسلمين.

كتبه

خالد بن عبد الرحمن الحصيّن

جامعة الإمام محمد بن سعود الإسلامية

كلية علوم الحاسب الآلي والمعلومات

ذو الحجة

الساعة العاشرة ليلة الإربعاء

1429\12\4

مقدمة في البرمجة

قبل البدء في البرمجة عليك أن تعرف أن البرمجة تعتمد على المنطق و المنطق فقط أي أن 1+1 =2 فليس هناك مجال اخر ربما لن تفهم ذالك الان ولكن ضعه في بالك حتى تبداء في كتابة البرامج. وكذالك أرجوا منك التمرس على كتابة البرامج لأن هذه الدورة لن تنفعك مالم تقوم بنفع نفسك, أعني أن هذه الدورة ستعطيك الأساسيات فقط ثم أنت يجب عليك أن تبحر في عالم البرمجة فلن نشرح إلا بعض الدوال الأساسية.أتوقع الأن أن جميعكم يقول (دع عنك الفلسفة) إذن لندخل في العميق و هيا لنشرح كود البداية:

كود:

```
#include<iostream>
using namespace std;
int main ()
{//first code
        cout<<"lutes  began student of imam university";
        return 0;
}
```

طيب لنبدأ الأن بسطر الأول و هو استدعاء لمكتبة (iostroame) وفيها بعض الدوال وحتى تفهم ذالك اصبر حتى نشرح دالة (cout) . أما عن (using name space std) فتستعمل لتضمين دوال (iostream) ثم يأتي بعد ذالك (int main) و هذا هو جسم البرنامج أي أن الشغل و البرنامج يأتي بعد ذالك .

أما ما بعد هذه العلامة// فهو خاص للمبرمج حيث أنه يضع شروحات البرنامج وشرح فوائد الدوال وتأتي فائدتها إذا بداءة بكتابة بعض البرامج التي تحتوي على دوال كثيرة فسوف تضيع إذا لم تضع هذه الشروحات. المهم هذا الذي فوق سيكرر في كل برنامج فأريد منك التركيز على الذي بعد main في هذا البرنامج استعملنا دالة واحدة و هي (cout) وتستخدم للإخراج. (cout<<)طبعا هذه الأسهم مهمة فهيه تأتي ملازمة ل (cout)ربما تسأل الأن مافائدة (cout) ,أقول لك فائدتها هو اخراج مايأتي بعد الأسهم يعني أن هذا البرنامج سوف يخرج عبارة (lutes began student of imam university) ومعنى العبارة: لنبداء طلاب وطالبات جامعة الإمام. طبعا علامة التنصيص فائدتها إخراج ما هو داخل العلامة وسنأخذ تفصيل عن ذالك في المدخلات و المخرجات. أما الأن فإن بودي أن أعود إلى دالة (iostream)جرب أن تحذفها وانضر هل سيعمل البرنامج أم لا طبعا البرنامج لن يعمل وسيعطيك تنبيه بأن دالة (cout) غير معروفة

وكذالك لو حذفت دالة (using namespace std) سيعطيك نفس المشكلة.
وبالنسبة للفاصلة المنقوطة (؛) فهذا يكون بعد الأوامر البرمجية وستعتاد ليها في المستقبل.
أما عن (getchar) فوظيفتها هو إبقاء البرنامج يعمل حتى تضغط على زر (enter) وجرب حذفها ستجد أن البرنامج يعمل ثم يخرج من البرنامج بسرعة.
طبعا هناك بعض العمليات الرياضية التي يجب أن تعرفها في ال ++c وهي:

العملية	الكود
الضرب	*
القسمة	/
الجمع	+
الطرح	-
باقي القسمة	%

طبعا الأولوية تأتي لما بداخل الأقواس ثم الضرب و القسمة ثم الجمع و الطرح أعتقد أن هذا واضح لذا لن أقوم بشرحه.

المتغيرات:
وهي عناصر تحتوي على قيم يحددها المستخدم مثل المتغيرات في الرياضيات.
طبعا هذه القيم يحددها المبرمج هل هو عدد صحيح أم عشري أم أنه حرف ، فلو فرضنا أنك كتبت برنامج يحتوي على متغيرين ولنفرض أنهما x و y وقد عرّفت أن x و y هما متغيران صحيحان وأعني بصحيحان أنهما لايقبلان الأعداد العشرية فلو أسند قيمة ما مثل 5,8890 إلى المتغير x فإنه سيأخذ الرقم 5 وسيتجاهل الباقي.
طبعا طريقة الإعلان عن المتغيرات هي كالتالي:
كود:

```
Int x,y;
```

وقد وضعت كلمة int ليعلم البرنامج أن x و y عددان صحيحان أما لو أردت أن اعرّفها على أنها عدد عشري فسوف أقوم بإستبدال كلمة int ب float أو ب double وهذا جدول يبين لك أهم الإعلانات للمتغيرات:

الإعلان	فائدته
Int	عدد صحيح
Float	عدد عشري
Double	عدد عشري ولكنه يقبل الأعداد الطويلة أي أن الأعداد الكبيرة التي لا يتحملها ال float
Char	يستخدم لتعريف الأحرف وكذاللك الأرقام

أما طريقة الإسناد فهي كالتالي :

كود:

```
X=5;
```

وتستطيع أن تسند القيمة مع الإعلان فتكون كذا:

كود:

```
Int x=5;
```

بل تستطيع أن تسند متغيرات لمتغير ، مثل هذا المتغير الذي يحوي متغيران وهو المتغير sumالذي يحوي قيمتين x +y ويكون الإعلان عنه بهذه الطريقة:

كود:

```
Sum=x+y;
```

فهذا المتغير يحتوي على المتغير x زائدا عليها المتغير y .
وتستطيع أن تجعل المستخدم يدخل قيم المتغيرات عبر ادالة (cin) والذي سنأخذه بعد قليل في المدخلات والمخرجات.
وإليك هذا المثال وهو برنامج يقوم بضرب قيمة متغيرين ويخرج الناتج:

كود:

```
#include<iostream>

using namespace std;
int main ()
{
        int x,y,sum;
        x=5;
        y=2;
        sum=x+y;
        cout<<x*y<<endl<<sum;
        getchar ();
        getchar ();
        return 0;

}
```

طبعا لن أشرح مافوق (main) لإنه وأضح وسيتكرر معنا ولنبدأ بما بعد (main) في البداية كما ترون عرفت أن هناك ثلاث متغيرات وهي (x,y,sum) طبعا نفصل بين المتغيرات بفاصلة ثم قمت بإسناد قيمة 5 ل (x)وقيمة 2 ل (y)وأسند جمع (x) و (y) ل (sum) ثم أخرجت قيمة ضرب x و y وبعد ذالك أخرجت قيمة (sum) طبعا كيفية الإخراج سنأخذه بعد قليل في المدخلات والمخرجات.

المدخلات والمخرجات:

المدخلات:

وهي دالة تستعمل لإدخال البيانات وتكتب هكذا (cin>>) فكما ترا القوس معاكس لقوس (cout<<)وتستطيع أن تسند القيم عبر هذه الدالة فكما رأيتم في درس المتغيرات الذي أسند القيم للمتغيرات هو المبرمج نفسه ولكن كيف أجعل المستخدم هو الذي يسند القيم ؟. أقول لك بعد أن تعرّف المتغير تضع بعدها دالة (cin) وهذه هي الطريقة:

كود:

```
Int x;
Cin>>x:
```

ولكن لو كان لدينا أكثر من متغير فكيف أجعل المستخدم يسند هذه القيم؟ أقول استعمل نفس الطريقة وضع بين متغير وأخر هذا السهم >> وانضر إلى هذا:

كود:

```
Int x,y,z;
Cin>>x>>y>>z;
```

وهذه طريقة اخرا:

كود:

```
Cin>>x;
Cin>>y;
Cin>>z;
```

وسنأخذ مثال على ذالك بعد أن نشرح المخرجات.

المخرجات:

وهي عكس المدخلات فهي تستعمل للإخراج وتكتب (cout<<) وأستطيع أن أخرج قيم المتغيرات عبر كتابة اسم المتغير بعد هذه الأسهم << وتتم بهذه الطريقة:

كود:

```
Cout<<x;
```

أما لو أردت أن تخرج أكثر من متغير فستكون بهذه الطريقة:

كود:

```
Cout<<x<<y<<z;
```

أو هذه الطريقة:

كود:

```
Cout<<x;
Cout<<y;
Cout<<z;
```

ولإنهاء السطر تستعمل الأمر endl وتكون بهذه الطريقة:

كود:

```
cout<<x<<endl;
```

وتستطيع أن تخرج جملة ما عبر وضعها بين علامة التنصيص مثل مثال كود البداية ومثل هذا المثال وهو نفس مثال المتغيرات ولكنه معدل طبعا هذا المثال سيكون مثال لدرس المدخلات والمخرجات كاملا:

كود:

```
#include<iostream>

using namespace std;
int main ()
{
        int x,y,sum;
        cout<<"enter tow number : ";
        cin>>x>>y;
        sum=x+y;
        cout<<"the number = "<<x*y<<endl;
        cout<<"the sum is : "<<sum<<endl;
        getchar ();
        getchar ();
        return 0;

}
```

طبعا شرح هذه الأكواد سأجعلها مشاركة والأن إلى الواجب:

س1: أكتب برنامج يطلب من المستخدم إدخال خمس أرقام ثم يقوم البرنامج بحساب المجموع والمتوسط لهذه الأرقام.

س2: اكتب برنامج يطلب من المستخدم إدخال خمس أرقام ثم يقوم البرنامج بوضع كل رقم في سطر.

تلميح للسؤال الثاني:تذكر أن int لايقبل إلا بالأعداد الصحيحة.

طبعا سأضع درسين خلال هذا الأسبوع و هذه الدرسين هي نهاية الفصل الأول. وسنناقش الدرس خلال هذا الأسبوع وسنستلم الواجبات .

*شرح سؤال المشاركة:

شرح سؤال المشاركة من الأخ هتلرر:

```
#include<iostream>

using namespace std;
int main ()
{
        int x,y,sum;
        cout<<"enter tow number : ";
        cin>>x>>y;
        sum=x+y;
        cout<<"the number = "<<x*y<<endl;
```

```
        cout<<"the sum is : "<<sum<<endl;
        getchar ();
        getchar ();
        return 0;

}
```

كود:

```
int x,y,sum;
```

هنا تم التعريف عن 3 متغيرات من نوع(int) اعداد صحيحة

كود:

```
        cout<<"enter tow number : ";
```

هنا طلب من المستخدم ادخال الرقمين المراد اجرا العملية عليمة ونفصل بين الرقمين بعملية الفراغ ( )space) ما ادري السبلينق صحيح والا لا .....

كود:

```
sum=x+y;
```

تم اسناد عملية الجمع الى المتغير sum

كود:

```
cout<<"the number = "<<x*y<<endl;
```

هنا اخراج ضرب العددين

كود:

```
cout<<"the sum is : "<<sum<<endl;
```

هنا تم اخراج قيمة المتغير(sum)وهوجمع العددين.

كود:

```
getchar ();
getchar ();
```

اعتقد انها لأيقاف تنفيذ البرنامج.

شرح الأخت أحلى روح:

كود:

```
#include<iostream>
```

هنا استدعاء لدوال (I\O) الإدخال والإخراج.

كود:

```
using namespace std;
```

تجعل المتغيرات متاحه للأستخدام في أي مكان في الكود.

كود:

```
int main ()
```

مثل ماقال أخ اباسليمان جسم البرنامج.

كود:

```
{
```

القوس المتعرج اعلان عن بداية البرنامج.

كود:

```
int x,y,sum;
```

عرَفت 3 متغيرات(x,y,sum) من النوع int.

كود:

```
cout<<"enter tow number : ";
```

جملة اخراج تطلب من المستخدم أن يدخل رقمان.

كود:

```
cin>>x>>y;
```

جملة ادخال يتم فيها ادخال المتغيران.(x,y)

كود:

```
sum=x+y;
```

هنا تم جمع المتغيرين المدخلين من المستخدم وحفظهما (اسنادهما) الى المتغير. sum

كود:

```
cout<<"the number = "<<x*y<<endl;
```

جملة اخراج ويتم فيها ضرب العددين المدخلين من المستخدم.

كود:

```
cout<<"the sum is : "<<sum<<endl;
```

جملة اخراج ويتم فيها اخراج قيمة. sum

كود:

```
getchar ();
 getchar ();
```

كود:

```
return 0;
}
```

و هنا لإنهاء البرنامج مع القوس المتعرج المغلق.

بسم الله الرحمن الرحيم
درسنا اليوم هو دالة الشرط . (if) و هو آخر درس في الفصل الأول لأن درس (swich) سنرجئه إلى الفصل الثاني.

قبل أن ابداء بدرس دالة الشرط أحب أن اضع هذا الجدول الذي يحوي على ادوات المقارنة:

أداة الربط	فائدته
==	يساوي
!=	لايساوي
>	أكبر من
<	أصغر من
>=	أكبر من أو يساوي
<=	أصغر من أويساوي

دالة الشرط: if
و هي من أهم الدوال المستخدمة في اللغة وطريقتها أن يتحقق البرنامج من تحقق الشرط فإذا تحقق الشرط عمل بما بعد الشرط.
فكأنك تقول للمترجم (إذا تحقق الشرط فاعمل كذا) ويكتب الكود بهذه الطريقة:

كود:

```
If  (statement);
```

طبعا (statement) هي جملة الشرط.
وبالمثال يتضح المقال:
كود:

```
#include<iostream>

using namespace std;
int main ()
{
        int x,y;
        cout<<"enter tow number : ";
        cin>>x>>y;
        if (x>y)
        cout<<"the largest is : "<<x<<endl;
        system("pause");
        return 0;

}
```

طبعا هذا المثال سوف يقارن بين قيمة x و y وإذا كان قيمة x أكبر من y فسوف يخرج قيمة x أما لو كان العكس فسوف يخرج من البرنامج ولكن يخرج شيئا.

ولكن نحن لانريد أن يخرج من البرنامج نريده مثلا أن يحسب أكبر قيمة فما العمل؟
أقول هناك دالة أخرى وهي عكس دالة if وهي دالة (else) وعملها عكس عمل (if) فكأنك تقول للمترجم (إذا لم يتحقق الشرط فاعمل كذا) ويكتب الكود بهذه الطريقة
كود:

```
else
```

وتستطيع أن تضيف ل (else)جملة شرط فيصبح الكود كذا:
كود:

```
else  if (statemint)
```

وانضر إلى هذا المثال وهو مثل المثال السابق ولكن قمنا بإضافة دالة (else) فأصبح البرنامج يطبع أكبر قيمة
كود:

```
#include<iostream>

using namespace std;
int main ()
{
        int x,y;
        cout<<"enter tow number : ";
        cin>>x>>y;
        if (x>y)
      cout<<"the largest is"<<x<<endl;
        else
    cout<<"the largest is :"<<y<<endl;
        system("pause");
        return 0;

}
```

الأن البرنامج سيقارن بين x و y وسيطبع أكبر قيمة.
طيب لو أردت من البرنامج أن يقارن بين ثلاثة أرقام ثم يقوم بإخراج أوسط عدد فكيف تعمل ذالك؟
في الحقيقة هناك أدوات تستعمل لمثل هذه الأشياء وهي أدوات الربط وهي كالتالي:

أدوات الربط	فائدته
And	&& ومعناها (و)

|| ومعناها (أو)	Or

بالنسبة للمقارنة AND الأصل نستعمل معها الرمز .... && وكذلك المقارنة OR الأصل نستعمل معها الرمز.... ||

أما إذا استعملنا الرمز & فهذا يرمز للعملية المنطقية .... AND مثل : 4&5
وكذلك الرمز | فهو يرمز للعملية المنطقية ..... OR مثل: 4|5.
طبعا وظيفة && و هو أن يتحقق البرنامج من تحقق الشرطين جميعا فإذا تحققا عمل بمقتضى الشرط.
أما وظيفة || هو التحقق من عمل أحد الشرطين فإذا تحقق إحداها عمل بقتضى الشرط.
وإليك هذا المثال و هو برنامج يطلب من المستخدم إدخال ثلاثة ارقام ثم يقوم بطباعة أوسطها.
كود:

```
#include<iostream>

using namespace std;
int main ()
{
        int x,y,z;
        cout<<"enter 3 number : ";
        cin>>x>>y>>z;
        if  ((x>y&&x<z)||(x<y&&x>z))
     cout<<"the midel is : "<<x<<endl;
        else if((y<x&&y>z)||(y>x&&y<z))
     cout<<"the midel is :"<<y<<endl;
        else
     cout<<"the midel is :"<<z<<endl;
        system("pause");
        return 0;

}
```

فكما ترون في الدالة الأولى للشرط قارنت بين x و y و z وطلبت من البرنامج أن يتحقق من x هل هو أكبر من y وأصغر من z أو أصغر من y وأكبر من z. فإذا تحقق هذا الشرطين طبع قيمة x.
أما إذا لم يتحقق الشرط فسوف ينتقل لشرط الثاني والشرط الثاني نفس الأول ولكن الفرق أنه سيطبع قيمة y.
أما إذا لم يتحقق أي من الشرطين فسوف يقوم بطباعة قيمة z.
و هذا مثال اخر على ادات الربط or و هو برنامج يطلب من المستخدم إدخال درجته ثم يطبع له عبارة " pass" إذا كانت درجته أكبر من 60 أما إذا كانت أصغر من 60 فسوف يطبع عبارة "fail"أي راسب، وإذا أدخل المستخدم درجة أكبر من 100 أو أصغر من الصفر فسوف يطبع عبارة "wrong in put" وتعني (مدخل خاطئ).
كود:

```
#include<iostream>
using namespace std;
int main ()
```

```
{//program of dgree
      int mark;
      cout<<"enter mark : ";
      cin>>mark;
      if (mark<0||mark>100)
      cout<<"wrong number"<<endl;
      else
      if (mark>60)
      cout<<"pass"<<endl;
      else
  cout<<"fial"<<endl;
 system("pause");
        return 0;

}
```

الأن سؤال المشاركة اشرح المثال الأخير.

أما سؤال الواجب فسيكون كما قال أخوي البرقاوي سؤال واجب إجباري وآخر اختياري:

س1 اكتب برنامج يطلب من المستخدم إدخال درجته ثم يقوم البرنامج بطباعة (a,b,c,d,f) وذالك حسب الدرجة المعطاة

س2 اكتب برنامج يطلب من المستخدم إدخال ثلاثة أرقام ثم يقوم بإخراج أكبر عدد وأصغر عدد.

أما السؤال الثالث وهو (اختياري)

س2 اكتب برنامج يطلب من المستخدم إدخال عدد صحيح ثم يقوم البرنامج بحساب العدد هل هو زوجي أو فردي فإذا كان زوجي طبع كلمة "even"أما إذا كان فردي فيطبع كلمة "odd" .

*شرح سؤال المشاركة :

شرح المثال للأخت أحلى روح:

راح ابدأ اشرح من بعد المين

كود:

```
int mark;
```

عرَفنا متغير من النوع int.

كود:

```
cout<<"enter mark : ";
```

طلب من المستخدم ادخال العلامه.

كود:

```
cin>>mark;
```

جملة ادخال يدخل فيها المستخدم العلامه.

كود:

```
if (mark<0||mark>100)
```

داله شرطيه (if) بقوم فيها البرنامج بمقارنة العلامه إذا كانت اصغر من الصفر أو أكبر من 100 راح يطبع الجمله التاليه

كود:

```
cout<<"wrong number"<<endl;
```

جملة اخراج تخبر بأن العدد المدخل خاطئ

كود:

```
else
```

تنفي عمل if يعني إذا كانت العلامه المدخله ليست أصغر من الصفر و لا بأكبر من 100.

كود:

```
if (mark>60)
cout<<"pass"<<endl;
      else
 cout<<"fial"<<endl;
```

هنا جملة if شرطيه اخرى تقارن هل العلامه المدخله أكبر من 60 إذا كان الكلام هذا صحيح راح تطبع pass وإلا أي إذا لم يتحقق الشرط فسيطبع fial.

والباقي مشروح في الدرس السابق....
بالتوفيق للجميع.....

الفصل الثاني

الحلقات التكرارية:

تعريفها:

الحلقات التكرارية اسم على مسمى فأنت تجعل البرنامج يكرر فعل ما حتى ينتهي الشرط الذي وضعته.

دالة: for

وهذه الدالة لاتستخدم إلا بعدّاد ولها وظائف عديدة منها إدخال القيم والبحث عن القيم طبعا سوف نستخدمها كثيرا في المصفوفات .

وطريقة كتابة الكود تكون بهذه الطريقة:

كود:

```
for(int i=0;i<=n;i++){
الأوامر البرمجية
}
```

قبل الشرح يجب أن تعلم أن في لغة c++ إذا وضعنا أمام المتغير (++) فسيزداد المتغير واحد، أعني لو كان المتغير يساوي 3 ثم وضعنا أمام المتغير (++) فسيكون 4 وإذا أردنا أن نزيد المتغير أكثر من واحد فنضع إشارة (n+) طبعا (n) معناه أي عدد يدخله المبرمج.
طبعا سأشرح معنى الأشيائ التي بين القوسين بعد هذا المثال.
و هو برنامج يطبع الأعداد من 0 إلى 10.
كود:

```
#include <iostream>
 using namespace std;

 int main()
 {
       for(int i=0;i<=10;i++)
       {
                   cout<<i<<endl;
       }
       getchar ();
       getchar ();
       return 0;
 }
```

في البداية عرفنا متغير وهو i على أنه عدد صحيح وساويناه بالصفر.
ثم وضعنا شرط الحلقة و هو أن تنتهي الحلقة بعد أن تكون i أصغرمن أو يساوي. 10
وبعدها وضعنا العداد.
الأن البرنامج سوف يعمل بأمر البرمجي الذي يأتي بعد القوس المنبعج({)
وقد وضعنا أمر إخراج لقيمة. i
البرنامج سوف يخرج في البداية قيمة i وهي الصفر ثم سيتزداد i واحد .
بعد ذالك سيتأكد البرنامج هل i أضغر من أو يساوي 10 ؟
إذا لم يكن يساوي أو أصغر من 10 سيستمر البرنامج في إخراج قيمة i وسيزداد i واحدا بعد كل دورة.
طبعا هناك دالة مشهورة تستخدم في دالة for وفي غيرها من الدوال
و هي دالةbreak
وتستخدم للخروج من حلقة for قبل إنهاء الشرط.
وانضر إلى هذا المثال و هو نفس المثال السابق وقد عدلنا عليه حيث سيخرج الأعداد من الصفر إلى الخمسة.
حيث أنه إذا وصل إلى خمسة خرج من الحلقة:
كود:

```
#include <iostream>
 using namespace std;

 int main()
```

```
{
        for(int i=0;i<=10;i++)
        {
                        cout<<i<<endl;
                        if(i==5)
                        break;
        }
        system("pause");
        return 0;

}
```

هناك كلمة مهمة في دوال التكرار وخاصة .... for وهي كلمة .... continue فائدة هذه الكلمة أنها تتجاهل الحالة المحددة وتتابع دورة التكرارمثل:

كود:

```
for(int i=0;i<=10;i++){
            if(i==5)
                    continue;
            cout<<i<<endl;
            }
```

هنا في هذا المثال: البرنامج سوف يطبع من 0 إلى 10 باستثناء رقم 5... لأنه إذا وصل إلى رقم 5 ينفذ أمر continue الذي يقوم بتجاهل ما أسفله من أوامر أو كود ويتابع دروة التكرار .....

هذا بشأن ال. for
طيب ماذا لو أردت من البرنامج أن يستمر في الحلقة من دون وضع عدّاد.
مثلا أخونا some bodey سألني عن إمكانية جعل البرنامج يستمر من دون أن أخرج أعني كل البرامج التي أشتغلنا عليها سابقا كان يخرج لك القيمة ثم يخرج من البرنامج نحن لانريد منه أن يخرج نريد منه أن يستمر، وأن أضع شرطا للخروج فكيف أفعل ذالك؟
هنا أقول لك يجب أن تستخدم الدلة. while
دالة: while
والمعنى الحرفي والفعلي لكلمة while هو (مادام) فكأنك تقول للبرنامج (مادام هذا الشرط موجود فأعمل بمقتضى البرنامج) طبعا له استعمالات عديد وتستطيع أن تجعله يعمل عمل for
.
ولا يشترط أن يكون هناك عداد.
طريقة كتابة الكود يكون كذا:

كود:

```
While (statemint){
```

```
الأوامر البرمجية.
}
```

طبعا (ststemint) معناه الشرط الذي يضعه المبرمج.
وانضر إلى هذا المثال وهو نفس المثال الذي في الفصل الأول وهو الواجب الذي حجلتموه
ولكن نجعل البرنامج يستمر حتى يدخل المستخدم (-1):
كود:

```
#include<iostream>
using namespace std;
int main ()
{
    int x,y,z,d,s,sum,ava;
    int optin;

    while (optin != -1)
      {
              cout<<"enter mumber1 :"<<endl;
              cin>>x;
              cout<<"enter number 2 :"<<endl;
               cin>>y;
              cout<<"enter mumber3 :"<<endl;
              cin>>z;
              cout<<"enter mumber4 :"<<endl;
              cin>>d;
              cout<<"enter mumber5 :"<<endl;
              cin>>s;
              sum=x+y+z+d+s;
              ava=sum/5;
              cout<<"the sum is :  "<<sum<<endl;
              cout<<"the ava is :  "<<ava<<endl;
              cout<<"enter any number to contino or -1 to
              quit :"<<endl;
              cin>>optin;
      }
    system("puase");
    return 0;

}
```

طبعا سأجعل شرح هذا المثال مشاركة.
هو في الحقيقة مثال مهم لذا يجب أن تشرحوه.

دالة التبديل:(switch) :

ومهمة هذه الدالة هي التبديل.
ويكتب الكود كذا:
كود:

```
 switch (المتغير){
الأوامر البرمجية
```

```
}
```

طبعا كما ترا يبداء بقوس منبعج وينتهي بقوس منبعج
هذا بشأن الإعلان عن المتغير الذي تريدون تبديله ولكن كيف تتم عملية التبديل؟
عبر إضافة دالة (case) ومعناه (في حالة) طبعا هذا الكلام النضري لن يكون واضحا من دون مثال.

وإليك هذا المثال وهو نفس المثال السابق ولكن أضفنا إليه دالة switch بحيث أن تضغط رقم 1 لحساب sum ورقم 2 لحساب ava ورقم 3 للخروج من البرنامج.
كود:

```
#include<iostream>
using namespace std;
int main ()
{
    int x,y,z,d,s,sum,ava;
    int optin;


    while (optin != 3)
      {
             cout<<"enter youer choice :" <<endl<<"1- for
              sum"<<endl
             <<"2- for ava"<<endl<<"3- forqout"<<endl;
             cin>>optin;

             switch (optin)
             {
                   if (optin==1)
                   case 1://sum

                         cout<<"enter mumber1 :"<<endl;
                         cin>>x;
                         cout<<"enter number 2 :"<<endl;
                         cin>>y;
                         cout<<"enter mumber3 :"<<endl;
                         cin>>z;
                         cout<<"enter mumber4 :"<<endl;
                         cin>>d;
                         cout<<"enter mumber5 :"<<endl;
                         cin>>s;
                         sum=x+y+z+d+s;
                         cout<<"the sum is : "
                          <<sum<<endl;
                   break;
                   case 2:
                         cout<<"enter mumber1 :"<<endl;
                         cin>>x;
                         cout<<"enter number 2 :"<<endl;
```

```
                    cin>>y;
                    cout<<"enter mumber3 :"<<endl;
                    cin>>z;
                    cout<<"enter mumber4 :"<<endl;
                    cin>>d;
                    cout<<"enter mumber5 :"<<endl;
                    cin>>s;
                    cout<<"the ava is :"
                     <<(x+y+z+d+s)/5<<endl;
               break;
          }
     }
    system("puase");
    return 0;

}
```

طبعا سأشرح هذا المثال ولكن قبل ذالك أظنكم أنكم قد لاحظتم وجود دالة break وهي مهمة في دالة التبديل لإنك بهذه دالة توقف عمل دالة التبديل.
كما ترا عرفنا متغير وهو (optin) ومعناه الخيار (طبعا هناك خطاء في الإملاء).
ثم استخدمنا دالة while لنقول للبرنامج مادام optin لا يساوي 3 فاستمر في البرنامج ثم طلبنا من المستخدم إدخال خياره.
بعد ذالك وضعنا دالة switch لنبدل المتغير . optin
و هنا بيت القصيد قلنا للبرنامج إذا كان optin تساوي 1 فاحسب لنا sum بعد ذالك قمنا بتبديل.
أي أننا قلنا للبرنامج في حالة أن ال optinيساوي 2 فاحسب لنا ال. ava
طيب هل أستطيع أن أكتب البرنامج من دون استخدام switch ؟
طبعا تستطيع بإستخدام دوال الشرط ولكن تخيل معي لو أن لديك عشر اختيارات ماذا ستفعل؟!
هل تقوم بوضع عشر دوال شرطية أم تقوم بفعل ذالك عبر دالة التبديل؟
تنبيه بسيط إلى أهمية أمر break في جميع حالات دالة .... switch فإذا لم تكتب هذا الأمر فإن البرنامج ينفذ الحالة المطلوبة وينفذ أيضا الحالات التي أسفلها حتى نهاية دالة..... switch

ومعنى أمر break هو" قف..."
هناك حالة افتراضية في دالة switch وهي معرفة باسم .... default وتنفذ هذه الحالة إذا لم يوافق الخيار المدخل من المستخدم أيا من الخيارات المتاحة في دالة .... switch مثل

كود:

```
int we;
    cin >> we;
    switch(we)
    {
        case 1:
             cout<<"first";
             break;
        case 2:
             cout << "second";
```

```
            break;
        default:
            cout << "wrong";
            break;
    }
```

فإذا أدخل المستخدم في هذا المثال رقم 3 مثلا ... هنا تنفذ حالة... default و هذا مثال آخر بسيط لن أقوم بشرحه أعتقد أنه واضح

كود:

```
#include <iostream>
using namespace std;



int main(){


    int mark;


    cout<<" pls enter the mark:";


    cin>>mark;
    if(mark < 0 || mark > 100)
            cout << " bad input \n";
    else {
    int markdiv10 = mark / 10 ;
    switch (markdiv10) {
       case 10:
       case 9 : cout << "A";
                break;
       case 8 : cout << "B";
                break;
       case 7 : cout << "C";
                break;
       case 6 : cout << "D";
                break;
       default : cout << "F";

    }
   }
    getchar ();
    getchar ();

    return 0;

}
```

أرجوا أن يكون الشرح واضحا.
وكذالك أرجوا أن تشرحوا المثال الذي طلبت منكم أن تشرحوه.
لإنه من المهم أن تفهموه . اذا أرجوا من جميع الأعضاء محاولة شرحه حتى ولو كان خطئا،
المهم أن تحاول.
والأن إلى سؤال الواجب:
س1: اكتب برنامج يطلب من المستخدم إدخال عشرة أرقام ثم يقوم البرنامج بحساب أكبر عدد وذالك بإستخدام دالة for.
س2: طلبت منك إحدى الشركات أن تصمم لها برنامجا بلغة c++ لحساب رواتب الموظفين.
و هذه الشركة لها عدة طرق لحساب رواتب الموظفين فهناك موظفين يستلمون رواتب أسبوعية وأخرون يستلمون نسبة 5,7% على كل قطعة يتم بيعها ، وآخرون يستلمون 250 ريال وعلى كل ساعة 20 ريال ولكن إذا كان عدد الساعات فوق 10 صار كل ساعة بـ40 ريال.
اكتب البرنامج بحيث يطلب من المستخدم الضغط على

1- لحساب رواتب الموظفين الأسبوعية
2- لحساب رواتب الموظفين بالقطعة
3- لحساب النوع الأخير من الموظفين

4- للخروج من البرنامج.
لن يكون في هذا الدرس سؤال اختياري ولكن حل ماتستطيع أن تحله وأرسله لي
وأسأل الله لي ولكم التوفيق في الدنيا والآخرة.

شرح المثال من الأخ the hero:

كود:

```
int x,y,z,d,s,sum,ava;
   int optin;
```

قمنا بتعريف 5 متغيرات من نوع int
وايضا قمنا بتعريف الجمع والمعدل والاختيار كمتغيرات من نوع int

كود:

```
while (optin != -1){
```

وضعنا شرط في داله while
و هو اذا كان الاختيار يساوي 1-
فانه يتوقف عمل البرنامج

كود:

```
cout<<"enter mumber1 :"<<endl;
    cin>>x;
```

هنا يتم ادخال العدد الاول

كود:

```
cout<<"enter number 2 :"<<endl;
          cin>>y;
```

ادخال العدد الثاني

كود:

```
cout<<"enter mumber3 :"<<endl;
    cin>>z;
```

ادخال العدد الثالث

كود:

```
cout<<"enter mumber4 :"<<endl;
    cin>>d;
```

ادخال العدد الرابع

كود:

```
cout<<"enter mumber5 :"<<endl;
    cin>>s;
```

واخيرا ، ادخال العدد الخامس>>>> اووووف مشوار

كود:

```
sum=x+y+z+d+s;
```

هذه المرحله يتم فيها جمع الاعداد

كود:

```
ava=sum/5;
```

هنا يتم حساب المعدل للاعداد

كود:

```
cout<<"the sum is : "<<sum<<endl;
```

هنا يتم طباعه المجموع

كود:

```
cout<<"the ava is : "<<ava<<endl
```

هنا تتم طباعه المعدل

كود:

```
cout<<"enter any number to contino or -1 to quit
:"<<endl;
 cin>>optin;}
```

هنا يطلب من المستخدم ادخال اي عدد لاكمال البرنامج او 1- للخروج منه والقوس المنبعج لاغلاق دالهwhile .

شرح المثال من الأخ somebody:

كود:

```
    int x,y,z,d,s,sum,ava;
    int optin
```

;

الإعلان عن 7 متغيرات ، خمسه خاصة بالأرقام ،

ومتغيرين خاصة بالمجموع والمتوسط ، ومتغير خاص بالإختيار.

كود:

```
 while (optin != -1){
```

هنا نشترط طالما المدخل لا يساوي(1-) ، يكمل
كود:

```
 cout<<"enter mumber1 :"<<endl;
    cin>>x;
          cout<<"enter number 2 :"<<endl;
          cin>>y;
            cout<<"enter mumber3 :"<<endl;
    cin>>z;
      cout<<"enter mumber4 :"<<endl;
    cin>>d;
      cout<<"enter mumber5 :"<<endl;
    cin>>s;
```

عمليات إسناد القيم ، وإدخال الأعداد بعد تحقيق الشرط.
كود:

```
    sum=x+y+z+d+s;
```

عرّفنا المجموع ،وأسندنا المتغيرات للمتغير sum

كود:

```
    ava=sum/5;
```

عرّفنا المتوسط.

كود:

```
    cout<<"the sum is : "<<sum<<endl;
    cout<<"the ava is : "<<ava<<endl;
```

عمليتين طباعة قيمة المجموع و المعدل.

كود:

```
     cout<<"enter any number to contino or -1 to
quit :"<<endl;
    cin>>optin;}
```

هنا يطلب من المستخدم إدخال أي عدد لإستمرار البرنامج ،

أو إدخال(1-) لإنهاء وإغلاق البرنامج.

كود:

```
 system("puase");
    return 0;}
```

إنتهينا !، وعطيكم العافيه' '

الفصل الثالث

الدوال function

في الحقيقة احترت كيف اعرِّف الدوال. ولكن مايجب أن تعرفه أن هذا الفصل سيكون نقطة تحول لحياتك البرمجية.

أستطيع أن أقول أن الدوال هي برامج وسط برنامج.

لا أتوقع منك أن تفهم التعريف الآن ولكن احفظ ذلك فإنه سيتضح لك قريبا .

يجب أن تعلم أن هناك نوعان من الدوال دوال ترجع بقيمة ودوال لا ترجع بقيمة. سنأخذ كل واحدة منهما على حدة.
ولكن قبل ذالك من المهم أن تعلم أن الدوال نكتبها خارج ال. (main)
تعلم أن الأوامر البرمجية تكتب عادة داخل main ولكن في هذا الفصل سنتعلم كتابته خارج mainثم سنستدعيه من داخل ال. main
أولا:الدوال التي ترجع بقيمة:
ويتم التعريف عنها بالإعلانات المعروفة (int, float,char...)وإليك الطريقة لكتابة الدالة:

كود:

```
int Name ()
```

طبعا كما ترى أعلنّا عن دالة ترجِع عدداً صحيحاً و name أقصد به اسم الدالة التي يسميها المبرمج.
مابين القوسين هي قيم (متغيرات) يضعها المبرمج.
وهي القيم التي أيضا تستقبلها الدالة.
وإليك طريقة الإستدعاء:
كود:

```
Name()
```

كما ترى تكون طريقة الإستدعاء بذكر اسم الدالة.
والتي بين القوسين هي القيم التي ترسلها الدالة main للدالة المستدعات.
دعك الأن من الكلام النظري ودعنا نأخذ هذا المثال وهي دالة سأسميها دالة (sum)دالة الجمع.وظيفتها جمع عددين:
كود:

```
#include <iostream>
using namespace std;
int sum(int x,int y)
{

       int  s= x+y;
       return s;
}

int main ()
{
         int x,y;
         cout<<"enter number : ";
         cin>>x>>y;
         cout<<"the sum is : "<<sum(x,y)<<endl;
         system("PAUSE");
     return 0;

}
```

سنركز الأن على دالة الجمع ثم سنعرج على دالة main لشرح طريقة الإستدعاء.

كود:

```
int sum(int x,int y)
{

      int   s= x+y;
    return s;

}
```

فكما ترا أعلنّا عن دالة ترجِع قيمة صحيحة ثم سمينا هذه الدالة ب.(sum)

و عرفنا بين القوسين متغيرين من نوع صحيح (طبعا ما بين القوسين هي القيم التي ستستقبلها الدالة.

بعد ذالك قمنا بتعريف متغير آخر صحيح وأسندنا له جمع x و. y

ثم قمنا بإرجاع قيمة s عبر الأمر البرمجي. (return)

الأن لنذهب إلى الدالة الرئيسية (main) لندرس طريقة الإستدعاء:

كود:

```
        int main (){
        int x,y;
        cout<<"enter number : ";
        cin>>x>>y;

        cout<<"the sum is : "<<sum(x,y)<<endl;

         system("PAUSE");
    return 0;}
```

كما ترى في البداية عرفنا متغيرين من نوع صحيح وأسميناه x و y ولا يشترط أن نسميه بنفس أسماء المتغيرات التي في دالة الجمع.

وأتوقع أنك تعرف الباقي لذا سأقوم بشرح طريقة الإستدعاء

كود:

```
sum(x,y)
```

طبعا كما قلنا سابقا طريقة الإستدعاء تكون بذكر اسم الدالة ووضع القيم التي تريد أن ترسلها بين القوسين.

هنا سيتم إرسال قيمة x التي يضعها المستخدم إلى دالة sum وسيتم إسنادها إلى المتغير x في تلك الدالة ونفس الشئ مع ال.(y)

بعد ذالك ستقوم دالة الsum بعملها وإرجاع قيمة. z

طيب ماذا لو قمت بتغير طفيف في الإستدعاء فأصبح بهذا الشكل:

كود:

```
sum(y,x)
```

الذي سوف يحصل هو أن ال yفي دالة ال mainستسند إلى ال xفي الدالة sum وال xفي دالة ال mainستسند إلى ال yفي دالة. sum
هذا مثال اخر فيه دالتان دالة sum ودالة: ava
كود:

```
#include <iostream>
using namespace std;
int sum(int x,int y)
{

      int    s= x+y;
      return s;
}
   double ava(double x,double y)
{
        double a=(x+y)/2;
        return a;
}

int main ()
{
        int x,y;
        cout<<"enter number : ";
        cin>>x>>y;
        cout<<"the sum is : "<<sum(x,y)<<endl;
        cout<<"the ava is : "<<ava(x,y)<<endl;
        system("PAUSE");
    return 0;

}
```

لن نشرح هذا المثال أتوقع أنه واضح.
قاعدة:
استقبال القيم للدالة يكون بين القوسين .... أما إرسال قيم في الدالة إلى خارج الدالة فيكون عن طريق.... return
ثانيا: الدوال التي لا ترجع بقيمة: طريقة الإعلان عنها تكون هكذا:
كود:

```
Void name ()
```

طبعا void معناه فارغ أي أنه لاترجع بقيمه.
والباقي نفس طريقة الدوال التي ترجع بقيمة.
طبعا الدالة void لاترجع بقيمة لذالك لاتحتوي على الأمر.(return)
طيب ما فائدة هذه الدوال التي لا ترجع بقيمة؟
أقول لك فائدتها تكون في حفظ الخطوات.
يعني لو عندك خطوات وأوامر برمجية تتكرر هل تقوم بكتابتها في كل مرة أمتقوم بوضعها في دالة وتستدعيها عند الحاجة .
سنأخذ الآن مثال على هذه الدوال وهذا المثال هو الواجب الإختياري (البرنامج الذي يطبع كلمة

(even)إذا كان زوجي و (odd) إذا كان فردي) طبعا البرنامج لا يرجع قيمة لذالك سيكون من نوع: void
كود:

```
#include<iostream>
using namespace std;
void math(int number)
{
     if(number%2==0)
    cout<<"even";
    else
    cout<<"odd";
}
int main()
{
    int number;
    cout<<"enter number:";
    cin>>number;
    math(number);
    getchar ();
    getchar ();
    return 0;

}
```

فكما ترى طريقة الإستدعاء سهلة مجرد كتابة اسم الدالة ووضع القيم المرسلة من ال mainإلى ال mathبين قوسين.
طبعا طريقة القوسين هذه أظن أنها معقدة قليلا لذا سأعدل على البرنامج السابق (حتي يتضح لك فائدة القوسين أكثر) وسأضع إدخال القيم داخل دالة math وبذالك لن نضع قيم بين قوسين:
كود:

```
#include<iostream>
using namespace std;
void math()
{
    int number;
    cout<<"enter number:";
    cin>>number;
    if(number%2==0)
    cout<<"even";
    else
    cout<<"odd";}
int main()
{
   math();
   getchar ();
   getchar ();
   return 0;

}
```

فكما ترى لم نضع قيم بين القوسين وذالك لأننا لن نرسل قيم من ال mainإلى ال math.
في الدوال : الأمر return ينهي عمل الدالة فمثلا في الكود التالي:

كود:

```
int sum (int n1 , int n2)
{
    int sum = n1+n2;
    return sum;
    cout << "END";
}
```

إذا استدعينا الدالة في الـ mainفلن تطبع النص "END" لأن الدالة تنتهي عند أمر return ....

وكذلك في الدوال من نوع .... void كالتالي:

كود:

```
void sum (int n1 , int n2)
{
    int sum = n1+n2;
    cou << sum;
    return;
    cout << "END";
}
```

هنا الدالة ستطبع المجموع ثم تنتهي عند أمر .... return لكن لاحظ هنا أمر return كتبناه بمفرده لأن الدالة void فهي لا ترجع أي قيمة...
أرجوا أن يكون الشرح واضح.
إخواني وأخواتي هذا الفصل في الحقيقة كبير جدا فيه تفاصيل ومعلومات كثيرة لم أذكرها (أعني هناك طريقة أخرى لكتابة الدوال وشيء من هذا القبيل)....
ولكن ماوضعته هنا هو مجرد الأساس والأساس فقط.

والأن إلى سؤال المشاركة:

كود:

```
#include <iostream>
using std::cout;
using std::cin;



int main()
{


int x, number = 1;
```

```
cout << "Enter integer: ";

cin >> x;


if (x > 0)
{
while (x > 0)
{
number *= x;
x--;
}

cout << "number: " << number;
getchar ();
getchar ();

return 0;
}
}
```

اشرح لي هذه الدالة واذكر لي فائدتها

والسؤال الثاني وهو مشاركة:

عدد فوائد الدوال.

والأن إلى سؤال الواجب

س1: اكتب برنامجا رياضيا بحيث يطلب البرنامج من المستخدم إدخال عددين ثم يقوم بسؤال المستخدم عن العملية المطلوبة بحيث تكون

1-للضرب

2-للقسمة

3-للجمع

4- للطرح

طبعا يكتب البرنامج على طريقة الدوال أي سيكون لديك 4 دوال.

الفصل القادم سيكون عن المصفوفات و هوا إنشالله آخر فصل.

كان من المفروض أن يكون الفصل الأخير هو ال classولكن في الحقيقة ليس لدي وقت لأكتب هذا الفصل لذا أرجوا منكم أن تعذروني .

وبالنسبة للفصل القادم هل تريدون أن أضع واجب أم مشروع.

شرح المثال من الأخت miss computer :

السلام عليكم ورحمة الله وبركاته..

هذه محاولتي للإجابة عن السؤال تبع المشاركة

كود:

```
int x, number = 1;
```

عرفنا عددين numberاعطيناه قيمة =1 والعدد xمتغير يؤخذ من اليوزر

كود:

```
cout << "Enter integer: ";

cin >> x;
```

طلبنا من اليوزر ادخال قيمة للرقم xصحيحه ..

كود:

```
if (x > 0)
{
while (x > 0)
{
number *= x;
x--;
}
```

اذا كانت القيمة المدخلة عدد موجب سندخل في الحلقة while وسيقوم البرنامج بالتحقق من قيمة x اذا كانت موجبة في كل مرة تدور فيها الحلقة
والمعادلة المكتوبة تعني ان نضرب ال numberاللي هو في الاساس =1 نضربه في x اللتي يحصل لها تناقص في كل مره بواحد(x--)
وراح يتوقف الدوران في الحلقة اذا كانت قيمه x سالبة تبعا للشرط(while (x > 0

كود:

```
cout << "number: " << number;
```

بعد ذلك نطبع النتيجة الاخيرة للرقم
****

الفائدة من البرنامج
استخراج المضروب...

بمعنى مثلا لو دخل اليوزر قيمة للـ xمثلا لنقل خمسة..
راح يكون الناتج
number=120
لان مضروب الخمسه = 1*2*3*4*5

الفصل الرابع
المصفوفات

هي مجموعة متغيرات مترابطة مع بعضها.

أعني بهذا التعريف تذكرون الواجبات والتمارين التي أخذناها من قبل بعضها كانت تحوي على أكثر من متغير ونعمل عليها نفس الحسابات. الآن لا حاجة لنا لنعرّف أكثر من متغير نستطيع أن نضع مصفوفة تحوي على هذه المتغيرات ثم نتعامل معها كمتغير واحد.

أعني تخيل أنه قد طلب منك كتابة برنامج خاص لمدرسة ما و هذا البرنامج يطلب من المستخدم إدخال درجة الطالب ورقمه واسمه ثم يقوم البرنامج بحفظ اسم الطالب ورقمه ودرجته، و هناك مائة طالب هل ستقوم بإعلان عن مائة متغير؟! أم ستكتفي بمصفوفة واحدة يحوي على مائة متغير.

طبعا لن أكتب البرنامج الآن سأكتبه بعد أن نأخذ طريقة كتابة الأكواد.

الآن سأشرح طريقة حفظ المتغيرات:

يجب أن تعلم أن الطريقة المتبعة في حفظ المتغيرات تكون بحفظ المتغير في خانة نعينه وتعطى في هذه الخانه عنوان للمتغير(انضر للجدول الآتي):

ar	13	6	20	70	45
	0	1	2	3	4

(هذا الجدول أخذ من كتاب ++c للمبتدئين للمؤلف: خليل الأمين عبدالجواد)
كما ترى هذه مصفوفة حجمها خمسة وأول عنصر هو (13) وموقعه هو الصفر والعنصر الثاني هو (16) وموقعه (1) وهكذا(تلاحظ أن المصفوفة بدأت بالصفر و هذا في ال ++c حيث إذا أعطيت مصفوفة ما حجم ما فسوف تبداء بالصفر يعني لو عرّفت مصفوفة حجمها عشرة فسوف تبداء المصفوفة من الصفر إلى 9
الآن سوف نأخذ كيف تتم كتابة الأكواد وكيف يتم الإعلان عن المصفوفة وكذالك كيف يتم إسناد القيم.
أما الإعلان عن المصفوفة فهذه طريقتها:

كود:

```
Int  ar[n];
```

كما ترى الإعلان عن المصفوفة يكون مثل الإعلان عن متغير ولكن الإختلاف يكون بهذه الأسهم) [n] معنى حرف ال n هو أي عدد يدخله المبرمج) فائدة هذه الأقواس هي تحديد حجم المصفوفة.
ويقصد بالحجم هو عدد العناصر المدخلة من قبل المستخدم والمبرمج.
طيب كيف يتم إسناد القيم للعناصر؟
تتم بهذه الطريقة:
كود:

```
Int ar[5]={1,2,3,4,5};
```

هنا اعلانا عن مصفوفة حجمها 5 وقد أسندنا إليها القيم.
في هذا الإعلان لو وضعت هذه الدالة:
كود:

```
Cout<<ar[0];
```

فسوف يخرج قيمة (1)، ولو وضعت مكان الصفر (3) فسوف يخرج البرنامج قيمة 4 وهكذا.
طبعا هناك أكثر من طريقة لإسناد القيم للعناصر.
أستطيع مثلا أن أسند القيم للعنصر الذي أريده بهذا الكود:
كود:

```
ar[3]={4};
```

هنا أصبح العنصر الثالث يحوي على العدد اربعة.
بل أستطيع أن أجعل المستخدم يسند القيم للعناصر بهذه الطريقة:
كود:

```
#include<iostream>
using namespace std;
int main ()
{
    int ar[6];
    for(int i=0;i<6;i++)
    {
            cout<<"enter number :\n";
            cin>>ar[i];
    }

    system("pause");
    return 0;

}
```

أما طريقة الإخراج يكون بإستبدال دالة cin ب cout.
السلاسل النصية:
في الحقيقة ليس لدي وقت لشرح عن السلاسل ولكن سأتكلم عنها بإختصار.
السلاسل النصّية اسم على مسمى فأنت تضع مصفوفة وتضع في كل عنصر حرف واحد ثم سيتم تجميع هذه الحروف على شكل كلمة انضر للكود التالي:
كود:

```
char ar[5] = {'h','e','l','o','\0'};
```

كما ترى في هذا الكود عرّفنا على مصفوفة تقبل الحروف ووضعنا كلمة (helo) ولكن قمنا يتقسيم الأحرف كل حرف في خانة.
و هذا امر متعب فتخيل أنك تطلب من المستخدم إدخال اسمه الرباعي فهل سيقوم يتقسيم الأحرف؟
في الحقيقة هناك دالة تريحك من عنى التقسيم و هذه الدالة هي دالة ال.string
دالة ال stringليست ضمن دوال ال iostreamلذالك وجب استدعائها.
وانضر إلى هذا البرنامج و هو البرنامج الذي أخبرتكم عنه في أول الفصل.
طبعا لن أجعله مائة طالب سأكتفي بعشر طلاب.
و هذا البرنامج يحتوي على ثلاث دوال.
دالة لكتابة اسم الطالب ورقمه وإعطاء درجته, والدالة الثانية للبحث عن رقم الطالب وإظهار درجته.
أما الدالة الأخيره فتطبع أرقام الطلاب حيث يطبع أرقام جميع الطلاب واسمائهم ودرجاتهم .
أما إذا كان رقم الطالب صفر فهذا يعني أن الطالب لم يسجل في النضام.
كود:

```
#include<iostream>
#include<string>
using namespace std;
```

```
void stu(int id[10],string name[10],int mark[10])
{
       int a;
       cout<<"give studint number : ";
       cin>>a;
       if(a>10)
       cout<<"roeng data";
       else
       {
              id[a]=a;
              cout<<"enter his name : ";
              cin>>name[a];
              cout<<"enter his mark : ";
              cin>>mark[a];
       }
}
void serch(int id[10],string name[10],int mark[10])
{
     int a;
     cout<<"enter the number of studint : ";
     cin>>a;
     cout<<"the name is : "<<name[a]<<endl;
     cout<<"the mark is : "<<mark[a]<<endl;
}
void print(int id[10],string name[10],int mark[10])
{
     cout<<"number \t"<<"name\t"<<"mark"<<endl;
     for (int i=1;i<10;i++)
     {
         cout<<id[i]<<"\t"<<name[i]
         <<"\t"<<mark[i]<<endl;
      }
}
int main ()
{
         int option;
         int id[10]={0,0,0,0,0,0,0,0,0,0};
         string name[10];
         int mark[10]={0,0,0,0,0,0,0,0,0,0};
          while (option!=4)
          {
             cout<<" 1-for studint info\n 2- for search
              \n 3- for print alist \n 4- for quet: ";
             cin>>option;
             switch (option)
             {
                    case 1:
                           stu(id,name,mark);
                    break;
                    case 2:
```

```
                    serch(id,name,mark);
                break;
                case 3:
                    print(id,name,mark);
                break;
            }
        }
        system("pause");
        return 0;
}
```

من خصائص المصفوفة من نوع int أو double أو float : أنك إذا أسندت قيمة إلى أول عنصر تلقائيا يسند 0 إلى العناصر الأخرى .... فبدلا من أن تكتب:

int id[10]={0,0,0,0,0,0,0,0,0,0};

تكتفي بكتابة:

int id[10] = {0};

وستسند إلى باقي العناصر 0....

كما ترى لم نحتج إلى عشر متغيرات بل مصفوفة واحدة لأرقام الطلاب وهذا ماعنيته في أول الفصل.وهذا مثال على المصفوفات والسلاسل النصّية وسيفيدك كثيرا في المشروع .

أرجوا أن يكون الشرح واضح.

لن أضع واجب في هذا الفصل بل سأضع مشروع بعد أن يضع أخوي برقاوي درس الكلاس .

لذا ارجوا إن كان هناك سؤال ما أو اقتراح أرجوا أن تطرحه بلا تردد.

وأخيرا أحب أن أشكر أخوي برقاوي الذي ساندني وساعدني في هذه الدورة والذي لو امتدحته إلي غدا فلن أوفيه حقه.

ولكني أسأل الله أن يجزيه عنا كل خير. وكذالك أرجوا منكم أن تدعوا له ولي ولجميع المشاركين بالخير والصلاح والتوفيق لنا ولوالدينا.

الكلاس: CLASS .....

أولا : لماذا نستخدم الكلاس ؟
نستخدم الكلاس لتنظيم الدوال والمتغيرات وترتيبها .... والكلاس أيضا هو أساس البرمجة الكائنية أي بواسطة الكائنات... (bjects)

طيب الآن كيف نعرف الكلاس ؟

أولا تكتب كلمة class ثم تكتب اسم الكلاس ثم تفتح قوس منبعج لبداية الكلاس ({) وفي نهاية الكلاس قوس منبعج مع فاصلة منقوطة(;{)
مثال:

كود:

```
class person
{

};
```

الآن تضيف الدوال والمتغيرات اللي تبغى .... مثال:

كود:

```
class person
{
     public:

     int age;
     int phone;
     string name;

     void display()
     {
          cout<<"name  : "<<name<<endl;
          cout<<"age  : "<<age<<endl;
          cout<<"phone  : "<<phone<<endl;
     }

};
```

طبعا أضفت ثلاثة متغيرات ودالة واحدة كمثال....
الجديد هنا كلمة : public هذه الكلمة تعني أن المتغيرات والدوال التي تكون تحتها يكون تصريحها عام .... يعني يمكن استعمال هذه الدوال أو المتغيرات في أي دالة خارج الكلاس ومنها دالة الـ mainوسنعرف ذلك لاحقا...

الآن سنعرف كيف نصل لمحتوى الكلاس ؟؟؟
إذا أردت الوصول إلى محتوى الكلاس تقوم أولا بتعريف كائن لهذا الكلاس (object) تصل بواسطته إلى محتوى الكلاس....
مثال:

كود:

```
#include <iostream>
#include <string>
using namespace std;

class person
{
     public:

     int age;
     int phone;
     string name;

     void display()
     {
          cout<<"name : "<<name<<endl;
          cout<<"age : "<<age<<endl;
          cout<<"phone : "<<phone<<endl;
     }

};

int main()
{
    person Barqawee;
    Barqawee.name = "Barqawee";
    Barqawee.age = 23;
    Barqawee.phone = 0555555555;
    Barqawee.display();

    system("pause");
    return 0;
}
```

في دالة الـ mainأولا عرفت كائن لكلاس person وسميته... Barqawee
ثم أسندت قيم لمتغيرات الكلاس مثل أي متغيرات أخرى لكن يكون استدعائها عنطريق الكائن
Barqawee ....
ثم استدعيت دالة... display

==
الآن سنرى النوع الثاني من التصاريح لمحتويات الكلاس و هو private ومعناه : الخاص....
نرى الآن كلاس person مع تصريح: private

كود:

```
class person
{
     private:
```

```
      int age;
      int phone;
      string name;

      public :

      void display()
      {
            cout<<"name : "<<name<<endl;
            cout<<"age : "<<age<<endl;
            cout<<"phone : "<<phone<<endl;
      }

};
```

أصبحت الآن المتغيرات الثلاث خاصة بالكلاس فقط وممنوع استدعائها خارج الكلاس ... فمثلا لو كتبنا الـ mainمثل المثال السابق:

كود:

```
int main()
{
    person Barqawee;
    Barqawee.name = "Barqawee";
    Barqawee.age = 23;
    Barqawee.phone = 0555555555;
    Barqawee.display();

    system("pause");
    return 0;
}
```

هنا سيصبح عندنا ثلاثة أخطاء في الترجمة .... وهي الأسطر التالية:

كود:

```
Barqawee.name = "Barqawee";
    Barqawee.age = 23;
    Barqawee.phone = 0555555555;
```

والخطأ هنا أننا استدعينا المتغيرات الثلاثة وهي.... private

طيب الأن كيف نسند قيم لهذه المتغيرات أو نستدعيها ؟؟؟
الحل يكون بإنشاء دوال تقوم بهذه المهام وتكون ... public كالتالي:

كود:

```
class person
{
      private:
```

```
	int age;
	int phone;
	string name;

	public :

	void setName(string n)
	{
		name = n;
	}

	string getName()
	{
		return name;
	}

	void setAge(int g)
	{
		age = g;
	}

	int getAge()
	{
		return age;
	}

	void setPhone(int p)
	{
		phone = p;
	}

	int getPhone()
	{
		return phone;
	}

	void display()
	{
		cout<<"name : "<<name<<endl;
		cout<<"age : "<<age<<endl;
		cout<<"phone : "<<phone<<endl;
	}

};
```

الآن نقدر نسند قيم إلى متغيرات الكلاس وأيضا نستدعيها ..... فتصبح الـ mainهكذا:

كود:

```
int main()
{
    person Barqawee;
    Barqawee.setName("Barqawee");
    Barqawee.setAge(23);
    Barqawee.setPhone (0555555555);
    Barqawee.display();

     system("pause");
     return 0;
}
```

أيضا ممكن نستدعي أحد قيم الكلاس عن طريق أحد الدوال التي أنشئناها ... مثلا:

كود:

```
int MyAge = Barqawee.getAge();
```

===================================

باقي معنا آخر نقطة مهمة وهي دالة البناء.... (constructor)
هذه الدالة تنفذ أول ما تعرف كائن لهذا الكلاس .... ويكون اسم دالة البناء نفس اسم الكلاس....
ولابد أن تكون.... public
نضيف الآن دالة بناء على الكلاس: person

كود:

```
class person
{
     private:

     int age;
     int phone;
     string name;

     public :

     person ()
     {
            age = 0;
            phone = 0;
            name = "";
            cout << "person has been built";
     }

     void setName(string n)
     {
          name = n;
     }
```

```
    string getName()
    {
            return name;
    }

    void setAge(int g)
    {
          age = g;
    }

    int getAge()
    {
            return age;
    }

    void setPhone(int p)
    {
          phone = p;
    }

    int getPhone()
    {
            return phone;
    }

    void display()
    {
          cout<<"name : "<<name<<endl;
          cout<<"age : "<<age<<endl;
          cout<<"phone : "<<phone<<endl;
    }

};
```

الآن أول ما تعرف كائن لهذا الكلاس في الـ mainوتطبق البرنامج تحصل إن دالة البناء تطبق تلقائيا مع تعريف الكائن....

====================

هذه أساسيات الكلاس .... أرجوا أن ينفع الله بها وأن يكون الدرس واضح....
و هناك أشياء أخرى مثل static و فصل الكلاس في ملفين و الدالة الهادمةdestructor
و غيرها .... إذا ودكم أطرحها ما عندي مانع...

=================

سؤال ذكاء : ماذا لو تم تعريف دالة البناء كـ private؟؟

============================

ملحق

حل الواجبات:

الفصل الأول

ج1: حل الأخ صامت:

```
#include <iostream>

using namespace std ;
int main ()

{

        double x,y,z,s,k,Sum ,Average ;
```

```
        cout<<"Enter 5 numbers\n";
        cin>>x>>y>>z>>s>>k ;
        Sum=x+y+z+s+k ;
        Average=Sum/5 ;
        cout<<" The sum is = " << Sum <<endl;
        cout<<" The average is = " << Average <<endl ;

        getchar () ;
        getchar () ;
        return 0 ;}
```

ج2:

```
#include<iostream>
using std::cout;
using std::cin;
using std::endl;
int main(){
    int number;
    cout<<"pleas enter 5 digits number: ";
    cin>>number;
    cout<<number/10000<<endl;
    cout<<(number%10000)/1000<<endl;
   cout <<(number%1000)/100<<endl;
    cout<<(number%100)/10<<endl;
    cout<<(number%10)<<endl;
    system("pause");

    system("pause");

return 0;}
```

ج1 من الأخت الطموح السامي:

```
#include<iostream>
using namespace std;


int main()
{

      int mark;

      cout<<"enter your mark"<<endl;
```

```
        cin>>mark;

        if(mark<0||mark>100)
                cout<<"wrong mark"<<endl;

        if(mark>90)
                cout<<"a"<<endl;

        if(mark>80)
                cout<<"b"<<endl;

        if(mark>70)
                cout<<"c"<<endl;

        if(mark>60)
                cout<<"d"<<endl;
        else
                cout<<"e"<<endl;


        return 0;
}
```

ج2 من الأخت الطموح السامي:

```
#include<iostream>
using namespace std;

int main()

{

        int num1,num2,num3,largest,smallest;

        cout<<"enter the three number"<<endl;

        cin>>num1>>num2>>num3;

        largest=num1;
        smallest=num1;


        if(num1>largest)
                largest=num1;

        if(num2>largest)
                largest=num2;

        if(num3>largest)
                largest=num3;
```

```
        if(num1<smallest)
                smallest=num1;

        if(num2<smallest)
                smallest=num2;

        if(num3<smallest)
                smallest=num3;

        cout<<"the largest is:"<<largest<<endl;
                cout<<"the smallest is:"<<smallest<<endl;
        return 0;}
```

ج3 من الأخ the hero :

```
#include<iostream>

using namespace std;

int main()
{
    int number ;
    cout << "please enter number:" << endl ;
    cin >> number ;

    if ( number%2==0 )
    cout << "the number is even" << endl ;

    else
    cout <<"the number is odd" << endl ;

    system ("pause");
    return 0 ;
```

حل الأخت الطموح السامي:

ج1

```
#include<iostream>
using namespace std;


int main()
{
        int num;
     int largest=0;


        for(int i=0;i<=10;i++)
        {

                cout<<"enter the number";
                cin>>num;

        if(num>largest)

        largest=num;}
        cout<<"the largest is:"<<largest<<endl;
        system("pause");

return 0;}
```

ج2 حل الأخت أحلى روح:

```
#include <iostream>
using namespace std;
int main()
{
        int option,choois;
        float salary,n,h,w;

        while(option!=4)
        {
          cout<<"Enter your choois : 1-For calculating the
salaries of staff weekly. 2-For calculating the salaries
of staff piecework. 3-For calculating the salaries of
other type of staff. 4-Out of the program:-   ";
          cin>>choois;
          switch(choois)
          {
                              case 1:
                                     cout<<"Enter salaries :-   ";
```

```
                                    cin>>salary;
                                    cout<<"salaries =
"<<salary<<endl;
                                    break;
                             case 2:
                                    cout<<"How Bit Rate :- ";
                                    cin>>n;
                                    cout<<(n/100)*5.7<<endl;
                                    break;
                             case 3:
                                    cout<<"How many hours you are
working :- ";
                                    cin>>h;
                                    if(h<=10)
                                    w=(h*20)+250;
                                    else
                                    w=(h*40)+250;
                                    cout<<w<<endl;
                                    break;
                             case 4:
                                    cout<<"exit\n";
                                    break;
                             default :
                                    cout<<" wrong\n";
          }
          cin>>option;

      }

     system("PAUSE");
     return 0;}
```

الفصل الثالث

ج1 من حل أخونا the hero

```
#include<iostream>
using namespace std;

 float sum ( float x , float y){
    float s=x+y;
    return s;}

     float subtract ( float x , float y){
         float s=x-y;
        return s;}

        float multiple ( float x , float y){
             float m=x*y;
            return m;}

             float divide (int x ,int y){
                float d=x/y;
                return d;}

                int main()
                {
                     float x,y;
                     int option;

                    cout<<"enter 2 numbers:\n"<<endl;
                    cin>>x>>y;

                    cout<<"select opration:\n"<<"1 for 
summation\n"<<"2 for substraction\n"<<"3 for 
multiplcation\n"<<"4 for division\n"<<endl;
                    cin>>option;

                    switch(option){
                                   case 1:{

cout<<sum(x,y)<<"\n";
                                        break;}

                                        case 2:{

cout<<subtract(x,y)<<"\n";
                                             break;}

                                             case 3:{
```

```
cout<<multiple(x,y)<<"\n";
                                                  break;}

                                                  case
4:{
cout<<divide(x,y)<<"\n";
break;}

default :
cout<<"error"<<endl;}

system ("pause");
return 0;
}cout<<"error"<<endl;}

system ("pause");
return 0;}
```

المشروع

```
include<iostream>
#include<string>
using namespace std;
void Reservation (int seat[10], string name[80] ,int
a)//Reservation
    {
      int clas;

                  cout<<"enter youer seat from 1 to10
"<<endl
                <<" if the stion is 1 that meen the seat
have been token:"<<endl;
                 cout<<"seat\t"<<"stion"<<endl;
                 for (int i=1;i<=10;i++){
                 cout<<i<<"\t"<<seat[i]<<endl;}

                cin>> a;
                seat[a]=1;
                cout<<"enter youer name:";
     cin>>name[a];
                  }

     void print (int seat[10] , string name[80],int
a){//print
     cout<<"enter youer seat:";
     cin>>a;
      cout<<"youer  seat is :"<<a<<"\t"<<"youer name is
:"<<name[a];}
     void cancel(int seat[10] , string name[80],int a){
          cout<<"enter youer seat";
          cin>>a;
          seat[a]=0;
          name[a]=" ";}//end print



                          int main (){//main
                              string name[80];
```

```
                            int
seat[11]={0,0,0,0,0,0,0,0,0,0,0};
                            int a,option;
                            while (option != 4){
 cout<<endl<<"1-Make Reservation"<<endl
 <<"2-Cancel Reservation"<<endl
 <<"3-Print Passengers list"<<endl
 <<"4-qouet"<<endl;

 cin>>option;
 switch (option){
        if (option==1)
        case 1:
                                Reservation(seat,name,a);
                                break ;
                                 case 3:
                                      print(seat,name,a);
                                      break;
                                      case 2:

cancel(seat,name,a);

                                 }}
                              system("pause");
                              return 0;}
```